CA Journal of Cultural Analytics

Stable Random Projection: Lightweight, General-Purpose Dimensionality Reduction for Digitized Libraries

Benjamin Schmidt

09.30.18

Peer-Reviewed By: David Mimno

Clusters: Data, Infrastructure

Article DOI: 10.22148/16.025

Dataverse DOI: 10.7910/DVN/UA0XBL & D20292750

PDF DOI: 10.31235/osf.io/36neu

Journal ISSN: 2371-4549

Cite: Benjamin Schmidt, "Stable random projection: lightweight, general-purpose dimensionality reduction for digitized libraries," Journal of Cultural Analytics. October 3, 2018. 10.31235/osf.io/36neu

Dimensionality reduction as humanities infrastructure

Digital libraries today distribute their contents in a way that limits the sort of work that can be done with them. Modern libraries are so large-often containing millions of books or articles-that the technical resources needed to work with them can be immense. Beginning researchers and students often cannot practically obtain more than a few thousand books at a time. Advanced researchers must use (often incomplete) metadata to decide which books are of interest for

their projects; and libraries themselves lack ways to make their full-text holdings easily discoverable by researchers or integrated with other collections.[1]

Humanists know that algorithms and infrastructure embody particular assumptions about the world. They are not "objective;" rather, they constrain some ways of thinking and promote others based on their creators and users assumptions about how the world should be. But though we know this, the emerging infrastructure for cultural analysis of texts often clashes with humanistic values and the field's desire for accessible research. A better infrastructure of algorithms and data would enable preliminary computational analysis and filtering of large digital libraries *before* researchers have to download terabytes of data or obtain rights agreements.

This article explores one way of making digital libraries more accessible: treating the algorithm used for dimensionality reduction as part of the social infrastructure of digital humanities, not as a task for the end researcher. It does so using a standard algorithm, *random projection*, which-despite a long history in applied mathematics and computer science-is only occasionally used in computational text analysis. For individual researchers, other methods work better. But, I argue, the method has other features that make it especially well-suited to shared textual work in the digital humanities. Combined with a trick based involving hash functions and random projection, it can reduce any text down to an easily-reproduced, arbitrary-length vector of numbers in a space that positions similar books close together, and dissimilar books far apart.

Treating dimensionality reduction as infrastructure means thinking of digital representations of books not just as 'machine-readable' texts, but as 'machine-read' texts: data that has already been partially digested by an algorithm. The choices we make for what this machine reading looks like shape the universe of possible research.

This article has three parts:

1. It describes the importance of dimensionality reduction; why it has generally been left as a task for the end researcher; and the systemic problems created by leaving it as a researcher-oriented task. I introduce the concept of a minimal, universal dimensionality reduction, which stands in contrast to existing methods which are poorly suited for large and/or multilingual

[1]I thank Peter Organisciak for several useful conversations about this article and for improvements to the underlying code base, and Andrew Goldstone and Scott Enderle for their comments on an earlier draft. An anonymous reviewer and Andrew Piper helped refine the argument for publication. I also gratefully acknowledge the support of a fellowship at the School of International and Public Affairs at Columbia University, under which much of this work were completed.

digital libraries. Such a reduction trades off some efficiency to produce embeddings of books that work in a wider variety of cases.

2. It describes a method for dimensionality reduction, **stable random projection**, which uses the technique of random projection in conjunction with hash functions to create a single low-dimensional space appropriate for a wide variety of texts and already-available features. Random projection is widely known in computer science as a passable, but not extraordinary, form of dimensionality reduction for texts. I argue here that it is particularly well suited to the circumstances of work in digital humanities compared to some of the similar methods in computer science.
3. It shows, through some examples, the uses of this space for supervised and unsupervised tasks on the full HathiTrust digital library of 13.6 million books. In particular, I explore how these features enable full-scale exploratory visualization of the full HathiTrust, and how relatively shallow neural networks with these features can allow classification on a wide variety of features encoded in library records. A vectorized version of the Hathi digital library, which is suitable for a much wider variety of tasks that can be explored in this paper, is included in the supplemental materials as a significant new data resource for any digital humanities research making use of library books.

Although the first two sections argue for a particular form of dimensionality reduction that sacrifices some classifier accuracy to better enable humanistic uses, many results and methods described in the third are possible under *any* dimensionality reduction or vectorization technique. This paper thus operates on two levels. On one, it introduces the idea of a lightweight, universal dimensionality reduction technique that can precomputed and easily distributed across platforms and languages, and proposes one candidate for such an algorithm. On the other, it starts to explore some of the research possibilities that may be possible with a minimal dimensionality reduction, with more traditional ones currently under development,[2] or with the deep-learning-based embeddings currently in vogue in machine learning. By providing a vectorized version of the HathiTrust library in the supplement, it makes it possible for others to begin exploring what might be possible with even more sophisticated vectorized representations of texts in coming years.

The third section focuses on classification and visualization in particular because the show clearly the advantages and opportunities of working with library-scale data. Visualization can make the scale and distribution of digital libraries acces-

[2] Peter Organisciak et al., "Access to Billions of Pages for Large-Scale Text Analysis." (iConference 2017, Wuhan, China, 2017).

sible in new ways. And classification based on large libraries can provide useful descriptions of documents even when metadata does not exist. For example: an architecture using these features and neural networks for classification can operate simultaneously in many different languages while correctly placing books into one of 225 Library of Congress subclassifications with quite high-68%-accuracy. This suggests a route for helping extend library metadata of all sorts into new domains and collections where it does not currently exist. At the same time, the ways and places that the classifier *fails* offer a window into understanding how historical taxonomies reflect the moment of their making. Classifier successes are not continuous across time, but instead reflect the history of library classifications themselves. The paper thus ends with a brief inquiry into how representations like these can help us explore the existing infrastructure for the organization of knowledge, as reflected in library practices.

Why Dimensionality Reduction Matters

In recent years, digital libraries including the HathiTrust library (c. 15m books) and JStor (c. 10m journal articles) have become increasingly committed to distributing "feature counts" as a first point of entry for various forms of textual analysis. They are less legally encumbered than full text, while still providing data that can be used for a wide variety of methods. These are among the most important parts of the emerging infrastructure for digital humanities work. Although they are usually adopted for legal reasons, they serve as an exemplar of the usefulness of machine-read texts in other ways; they are generally smaller than full text files, and by enforcing a single tokenization scheme help harmonize work by different researchers.

Feature counts, however, are only occasionally useful inputs in themselves into machine learning representations. They are both too large (the full Hathi feature counts contain more than 100 billion data points) and too irregular to easily be integrated into many standard clustering and classification algorithms. Further complicating matters is that the legal status of these feature counts themselves can be somewhat murky; Hathi, for example, took some time to release features on in-copyright works after publishing public-domain works in 2014, and JStor distributes feature counts only under restrictive licenses.[3]

[3] Boris Capitanu, Ted Underwood, Peter Organisciak, Timothy Cole, Maria Janina Sarol, J. Stephen Downie (2016). The HathiTrust Research Center Extracted Feature Dataset (1.0) [Dataset]. HathiTrust Research Center.

For computational purposes, feature counts are best understood as a representation of the **term-document matrix**, the standard abstraction of a text corpus to a "bag of words" representation. For a large text corpus like the Hathi Trust, the term-document matrix consists of millions of rows, each one representing a single book; and millions of columns, each representing a single word. At each point, the number of times an individual word is used in an individual book is stored. In theory, this matrix can be extremely large; the HathiTrust Bookworm browser, for example, which removes words that appear fewer than 50 times, has 13 million books and about 4 million distinct words. The naive approach to storing this as a rectangular term-document matrix would take about 150 terabytes (40 large consumer-grade hard drives) to store.[4]

In practice, feature counts are distributed in a sparse form that makes them considerably smaller (though still unwieldy) by not including word-document interactions with a count of 0. Still, a dense form is generally necessary for a wide variety of statistically techniques, from neural networks to logistic regression to k-nearest-neighbor algorithms. Some sort of **dimensionality reduction** is therefore a frequent step in analysis. Rather than requiring millions of columns, dimensionality reduction algorithms find ways to combine word counts together or to eliminate some entirely. In this new space, it is easy to apply the wide variety of statistical techniques designed for dense matrices.

This creates a largely unacknowledged need in digital research. While tokenization is on the verge of becoming a regular service provided by digital libraries, dimensionality reduction is not. With only feature counts, large-scale digital libraries are all but inaccessible. Dimensionality reduction is generally quite computationally expensive. It requires a great deal of processing power and physical storage to work with even a reduced set from a corpus like the Hathi Trust; this can make the "big data" side of digital humanities is almost entirely inaccessible without access to high-performance computing and extraordinary amounts of storage.

Standard dimensionality reduction

One reason that dimensionality reduction is left to researchers is that although there are a variety of techniques for dimensionality reduction widespread in the digital humanities, each has fundamental features that make it difficult to use

[4] Hathi+Bookworm

outside of a single research project. The simplest dimensionality reduction is to drop all but the most common words in set, as measured either by overall frequency or by the number of documents in which they appear. In both the digital humanities and computer science, scholars most frequently use "top-N" words as a good enough approximation of the textual footprint. It reduces the dimensions to a few hundred of the most common words in the corpus; this has produced what Maciej Eder has characterized as "endless discussions of how many frequent words or n-grams should be taken into account" for stylometry.[5]

The gold standard for dimensionality reduction are techniques that make use of co-occurrences in the term-document matrix such as latent semantic indexing and independent components analysis. More recent techniques such as semantic hashing can be even faster and more efficient at optimally organizing documents in various types of vector spaces designed especially for particular documents.[6]While individual researchers are wise to use these methods in their own work, they suffer two problems that make them problematic as a way for digital libraries and researchers to share dimensionally-reduced features for others to work with.

First, they are computationally complex, and can be difficult to perform on a very large corpus. Many require singular value decomposition (SVD) as an initial step. Dimensionality reduction on large textual datasets is computationally quite expensive. A set like the Hathi Trust books may consist of 10 million distinct tokens across 15 million individual books; SVD on a matrix with hundreds of trillions of entries is difficult to perform. This has led researchers to propose sampling techniques which could mitigate the difficulty at the expense of introducing random fluctuations.[7]

Second, it is difficult to project *out-of-domain* documents into the space from a standard projection. The greater the difference between out-of-domain documents and a reference corpus, the more problematic out-of-domain projection becomes. Features that are collinear in one set may not be in another: for instance, "bank" and "river" might be highly collinear in texts about geology, but

[5]Maciej Eder, "Visualization in Stylometry: Cluster Analysis Using Networks," *Digital Scholarship in the Humanities*, December 2, 2015, fqv061, doi:10.1093/llc/fqv061.

[6]Scott Deerwester et al., "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science* 41, no. 6 (September 1, 1990): 391-407, doi:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9; Ruslan Salakhutdinov and Geoffrey Hinton, "Semantic Hashing," *International Journal of Approximate Reasoning*, Special section on graphical models and information retrieval, 50, no. 7 (July 2009): 969-78, doi:10.1016/j.ijar.2008.11.006

[7]Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp, "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions," *arXiv:0909.4061 [Math]*, September 22, 2009.

quite different in texts about economic geography. Moreover, tokens not represented in the source corpus have no defined location whatsoever in the new space, meaning large amounts of useful information may be lost. Proper nouns (names of corporations, for example, or researchers) which are common and useful in one corpus may be not present at all in another. A list generated by the first eighty years of a scientific corpus will not have rules for new technical vocabulary that emerges in year eighty-one. Any dimensionality reduction using word counts will therefore display classical "algorithmic bias" towards vocabulary; it will assume that texts like those it sees often matter, and that those that are rare or nonexistent are unimportant.

This out-of-domain problem presents a particularly great problem with multilingual corpora. Alan Liu has spoken recently of the need to "solve the language problem" in digital humanities, in which algorithms like topic modeling only work on one language at a time.[8] In this case the most important problem is of language composition; most dimensionality reductions will inevitably privilege the top languages in a corpus.[9] The richness of features for any language will be directly proportional to its representation in the original set. For example: in a corpus of 95% English and 5% German-language text, most information in a dimensionality reduction will be specific to the English language. There will be no information retained at all for documents written in Spanish, except for words that happen to appear in one of the other languages. I emphasize emphasize multilingual data retention in this paper, because it is a case many researchers will be familiar with. It is, though, only the most striking example of the general case that an important but minority vocabulary would be lost using features or vocabularies built across a larger corpus.

Even when a new set of documents are of the same type as those in the initial reduction-for instance, if a researcher wants to add 10 newly discovered novels to an existing corpus-it can require significant computational resources to project new documents into the same space. To share the rules for transformation, an entire m x n matrix must shared, where m is vocabulary size and n is the desired number of dimensions. With a large corpus like Hathi, a reasonable set of choices might involve 100,000 words and 1,000 dimensions; in order to project a new document into this space, a researcher would need to download half a gigabyte of data, rendering it unusable for purposes like online web services. With more sophisticated algorithms like semantic hashing, precise out-of-domain ap-

[8] Alan Liu, "Varieties of Digital Humanities" (Modern language association 2018, New York City, 2018).

[9] Nick Thieberger, "What Remains to Be Done—Exposing Invisible Collections in the Other 7,000 Languages and Why It Is a DH Enterprise," *Digital Scholarship in the Humanities* 32, no. 2 (June 1, 2017): 423-34, doi:10.1093/llc/fqw006.

plication is impossible by design; only the originally-described documents have any position in the new space at all.

Minimal, universal dimensionality reduction

Most widely used techniques for dimensionality reduction, therefore, make out-of-domain projection quite difficult, or even impossible, in order to maximize the information conveyed through the reduction for the specific task at hand. Rather than optimizing for information storage, humanists and librarians may want their dimensionality reductions to prioritize something different: the ability to work on a wide variety of texts and in a wide variety of contexts. Such a dimensionality reduction would be more appropriate for distribution by a library than one specific to their particular corpus. I call it a minimal, universal dimensionality reduction because it would, ideally, do three things well.

1. It would **reduce dimensionality**: it will represents texts as a set of numbers in a way that significantly reduces their size, while preserving similarities and differences between them as far as possible. Any form of dimensionality reduction is extremely useful with texts: a 640-dimensional dimensional projects takes 2.56kb of space to describe a single book; the full HathiTrust corpus can be stored in about 30GB of data, compared to roughly 1,500 GB for counts of each individual word. A subset such as 140,000 works of fiction can be comfortably loaded into memory on a laptop.
2. It would operate **universally**: the reduction will not learn techniques for reduction from one corpus that are less appropriate in another, and the same space will be suitable to represent any text in any subject area or language. It is worth noting that universal linguistic applicability is distinct from something much harder: a cross-lingual projection in which, for instance, English and German texts about politics would be close to each other. In practice any minimal universal reduction will, almost certainly, group texts by linguistic similarity first and only later by style, subject matter, or any of the other features researchers are interested in.
3. It would operate **minimally**: the rules for reduction will not require a large lookup table or centralized registry of words, but can be represented in code alone. This means it could run in a web browser, or on lightweight hardware like that used by the group for minimal computing.[10]

[10] "Minimal Computing. Minimal Computing: A Working Group of GO::DH," 2017.

Such a minimal projection would allow techniques that build on dimensionality reduction to be more practical in new contexts. Information providers like the Hathi Trust and Jstor could distribute reduced features usable for initial research tasks at considerably lower size and less security risk than unigram counts. Currently, any reduced feature set would be limited in its usefulness because it would lock in the current state of the corpus. Unlike any learned reduction, a minimal projection is not constrained by the contents of the library and so can be useful for research on any sub-corpus an individual researcher might bring, including highly specialized vocabularies or uncommon languages.

The most significant benefit to individual researchers is that they do not have to perform dimensionality reduction themselves, which can be more computationally complex than actual analysis. Minimal reductions aenable exploratory data analysis in which almost any extant corpus to be read directly into memory on a personal computer in a reduced form, which can dramatically reduce the hardware requirements necessary to begin modeling sets of texts. But there are other possibilities that arise with scalability around a standard feature set. Researchers could distribute among themselves classificatory models that can be applied on any set of texts. For example, a model that estimates the prevalence of optical-character-recognitions misreadings in one corpus might be applied on another to determine its quality. Web portals can deploy an infrastructure where documents can be projected into a space before sent to a server, saving the need and risks (to privacy and copyright law) of transmitting a full document to the server.[11]

All of these infrastructural challenges are poorly met by conventional dimensionality reductions that work to maximize information retention rather than promote reuse.

Stable Random Projection: the method

Random projection offers a form of dimensionality reduction that comes close to meeting the criteria above. Random matrix theory has emerged in the past few decades as an useful alternative to more computationally complex forms of dimensionality reduction, finding use in a variety of fields from medicine to the creation of word embeddings.[12] As with other matrix-based dimensionality re-

[11] One useful recent example of the possibilities of this infrastructure is JStor's Text Analyzer

[12] Ella Bingham and Heikki Mannila, "Random Projection in Dimensionality Reduction: Applications to Image and Text Data," in *Proceedings of the Seventh ACM SIGKDD International Conference*

ductions, random projections can be thought of as multiplying together two matrices. In textual data, the first, D, might be the term-document matrix: a $d \times v$ matrix where d is the number of documents and v is the number of distinct tokens, with each entry $D_{i,j}$ corresponding to the number of times document i uses token j. Since there are many possible words, v is large (perhaps 100,000 to 1,000,000). The second matrix, T, is a transformation matrix of shape $v \times n$, where n is the number of dimensions in the reduced space (perhaps 100 to 1,000) and each entry $T_{i,j}$ gives the weight for word j in dimension i. The dot product of these two matrices, $D \bullet T$, yields a $d \times n$ matrix S, which is the projection of each document in D into the new n-dimensional space.

While methods like LSA carefully learn appropriate values for the transformation matrix , that project each word into an efficient space of reduced dimensionality, random projection, as the name implies, instead fills the transformation matrix with random values that have no relation to the original matrix. Perhaps surprisingly, while the meanings of the individual dimensions are random, the relationships of points to *each other* persist even after this randomization. One foundational finding in the literature, the Johnson-Lindenstrauss lemma, establishes that the lower dimensional projections produced by certain random distributions can come close to maintaining the relative distances between all the higher dimensional points.[13]

In short, each dimension of a randomly projected feature set generally contains some information about every one of the input dimensions; while each individual resulting features is intrinsically meaningless, in combination they allow a significant amount of the original data to be reconstructed. Initial work in random matrices projected each dimension according to a normal distribution; more recent work has established computationally simpler methods. For the purposes of this paper, an especially important finding is that the random matrix can be created purely by sampling randomly from the set [-1,1].[14]

on Knowledge Discovery and Data Mining (ACM, 2001), 245-50, Teija Seitola et al., "Random Projections in Reducing the Dimensionality of Climate Simulation Data," *Tellus A: Dynamic Meteorology and Oceanography* 66, no. 1 (December 1, 2014): 25274, doi:10.3402/tellusa.v66.25274, Haozhe Xie, Jie Li, and Hanqing Xue, "A Survey of Dimensionality Reduction Techniques Based on Random Projection," *arXiv:1706.04371 [Cs]*, June 14, 2017, Magnus Sahlgren, *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations Between Words in High-Dimensional Vector Spaces*, SICS Dissertation Series 44 (Stockholm: Dep. of Linguistics, Stockholm Univ., 2006).

[13] William B. Johnson and Joram Lindenstrauss, "Extensions of Lipschitz Mappings into a Hilbert Space," *Contemporary Mathematics* 26, no. 189 (1984): 1.

[14] Dimitris Achlioptas, "Database-Friendly Random Projections: Johnson-Lindenstrauss with Binary Coins," *Journal of Computer and System Sciences*, Special issue on PODS 2001, 66, no. 4 (June 2003): 671-87, doi:10.1016/S0022-0000(03)00025-4.

Random projection, it is worth emphasizing, is objectively worse at retaining information than methods like latent semantic indexing, principal components analysis, or independent components analysis. Since feature vectors in text tend to be highly collinear, a great deal of information can be saved by having similar words aligned in the same directions as each other. The established literature on random projection for textual data has thus tended to give less shrift to random projections as result.[15] The relatively inefficiency of random projection can be alleviated by subsequently using another reduction technique on the SRP features, such as principal components, before computation.

A standardized random matrix projection of textual data

Classical random projection comes close to being a universal dimensionality reduction, but not to being a minimal one. A truly random matrix would require generating a random array for every token and maintain it as a central resource. This makes creating new projections into the same space quite difficult; and to distribute the rules for projecting documents into a random projection space could take hundreds of megabytes. Some uses of random projection in the research literature, in fact, make use of the difficulty of reproducing random projections as a security feature to help keep data confidential.[16] In most digital library research, by contrast, reproducibility is quite important.

Instead of requiring a central registry, I use here a trick that makes it possible to materialize a quasi-random projection matrix for any set of strings that can be easily computed on any platform.[17] I call this "stable" random projection to emphasize that the same projections can be created across computer systems and languages.[18] Cryptographic hashes can provide a consistently reproducible

[15] Tang, "A Comparative Study of Dimension Reduction Techniques for Document Clustering Faculty of Computer Science," 2004.

[16] Devansh Arpit et al., "An Analysis of Random Projections in Cancelable Biometrics," *arXiv:1401.4489 [Cs, Stat]*, January 17, 2014, T. Bianchi, V. Bioglio, and E. Magli, "Analysis of One-Time Random Projections for Privacy Preserving Compressed Sensing," *IEEE Transactions on Information Forensics and Security* 11, no. 2 (February 2016): 313-27, doi:10.1109/TIFS.2015.2493982

[17] I am unaware of any other work using binary hashes this way as an input to low-dimensional random projection matrices; this method bears some relationship to the widely used "hashing trick" (discussed further below) which maps each word to a single location in a high-dimensional space.

[18] This phrase has also been used by Ping Li in a different context to describe random projections that provide stable estimates according to various distance metrics. Ping Li, "Estimators and Tail

quasi-random number generator for any token which is easily transformed into a random projection matrix.[19] The SHA-1 hashing algorithm transform a variable-length string to a fixed-length number. This is typically represented as a hexadecimal string: for instance, the SHA1 hash of the string "bank" is bdd240c8fe7174e6ac1cfdd5282de76eb7ad6815. Represented in binary, this is a 160-bit number beginning with the numbers 1011 1101. Achlioptas[20] established that a random sampling of the numbers [-1,1] is effective as a random projection matrix; SRP uses each element the SHA-1 hash to generate such a random matrix for any given token. The stable random projection of a token is defined as 1 if the corresponding bit in the token's SHA1 hash is 1, and -1 if it is zero. For example, the first 8 bits of the SHA-1 hash for bank are [1,0,1,1,1,1,0,1], so the projection of "bank" begins [1,-1,1,1,1,1,-1,1,…]. (Functioning implementations of the algorithm described here in Python, R, and Javascript are provided in the appendix.) To extend the projection beyond 160 dimensions, the same method is reused, but with the character _ added to the end of the string. (For example, the 480-dimensional projection of "bank" is the same as the 160-dimensional projects of the words "bank," "bank_," and "bank__" concatenated together.) The large number of dimensions ensures that no two words will have an identical projection; it took Google more than a year of computation on over 100 GPUs to discover a single SHA-1 collision in 2017.[21]

In formal notation: at any given position i, the SRP hashing function h of word w casts the corresponding bit of the SHA-1 function to the set [-1,1].

$$h(w)_i = sha1(w)_i * 2 - 1$$

To generalize to a full text, rather a single word, the most obvious method is to simply sum word counts. Given a document D, with distinct vocabulary of words w of length W, the hashing function h described above, and a set of word counts c where c_i is the number of times that w_i is used in a document, a preliminary function SRP can be represented in the following expression:

Bounds for Dimension Reduction in LA (0 ≤ α ≤ 2) Using Stable Random Projections," in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08 (Philadelphia, PA, USA: Society for Industrial; Applied Mathematics, 2008), 10-19.

[19] The choice of a hashing function is relatively unimportant; I choose SHA-1 because implementations are easily available in almost all programming languages. PUB FIPS, "180-1. Secure Hash Standard," *National Institute of Standards and Technology* 17 (1995): 45.

[20] Achlioptas, "Database-Friendly Random Projections."

[21] Marc Stevens et al., "Announcing the First SHA1 Collision. Google Online Security Blog," February 23, 2017.

$$SRP'(D)_i = \sum_{n=1}^{W} h(w_n)_i * c_n$$

Put less formally, the SRP projection of any individual document can be thought of as created in the following way.

1. Choose any number of dimensions, and preassign a "zero" score for as many dimensions are desired.
2. Starting with the first dimension, use the SHA-1 hash function to quasi-randomly designate each word that appears in the document as being *positive* or *negative* for this particular dimension.
3. For each word designated "positive" for this dimension, add its wordcount in the document to the score for this dimension
4. For each word designated "negative" for this dimension, subtract its word-count in the document from the score for this dimension.
5. Repeat 2-4 until all the dimensions are done.

The net result of this process is that each dimension contains some information about the word counts for every word; the dimension is marginally higher if the bit for that dimension's SHA-1 hash is 1, and marginally lower if the bit is 0. Each additional dimension makes it possible to more easily trace out the contributions of any individual word, while the overall scores for each dimension should be normally distributed with a mean of zero.

Text pre-processing

Although the algorithm described above takes "word counts" for granted, generating them requires a large number of interpretive choices. Matthew Denny and Arthur Spirling, in a useful recent article, identify seven different pre-processing steps frequently taken by researchers, that together yield 128 different tokenizations of any text.[22] I follow their taxonomy in describing SRP's tokenization algorithm. SRP aim at introducing the greatest regularization possible without introducing any rules based on a particular language. Thus the default implementation removes punctuation, lowercases all words, and replaces numeric digits

[22] Matthew Denny and Arthur Spirling, "Text Preprocessing for Unsupervised Learning: Why It Matters, When It Misleads, and What to Do About It," SSRN Scholarly Paper (Rochester, NY: Social Science Research Network, September 27, 2017).

with the '#' sign; but it does *not* stem (lemmatize) words, remove stopwords, or remove infrequent words, because those require language-specific rules. It also does not include bigrams or trigrams for a purely practical reason: some corpora (including Hathi) are only available to researchers as unigram counts.

Tokenization in natural language processing is best defined at the level of individual languages; any multilingual hashing scheme will be necessarily imperfect. I have chosen one that matches any continuous series of letter or digit characters as defined in the Unicode specification. The regular expression in the reference (python) implementation is \w+; for example, the string "Françoise doesn't have $100.00" is normalized and tokenized to ["françoise","doesn","t","have","###","##"].

Tradeoffs

There are a number of cases where random projections falls short of full "universality" as described above. While any Unicode text can be parsed with this regular expression, it should work best in languages where "words" and tokens are relatively synonymous. The most important is the handling of languages that do not lend themselves to a tokenization algorithm that relies on adjacent word-like characters. This can be seen clearly in the next section: the performance of the classifier is worst on languages like Thai, Chinese, and Urdu which may not use whitespace delimitation of characters. Chinese and Japanese perform better on classification scores, but the texts used here were pre-tokenized by the Hathi Trust Research Center; it is possible that performance on them would be similarly low if not the code could be trained on the raw text.

The problems with Chinese are especially important, but could be solved through some specialist intervention. Later version of the algorithm might be to change its treatment of multi-character words. "Words" in the CJK Unified Ideographs unicode block, for example, could be parsed as a set of two-character overlapping shingles if they contain more than a small number (e.g., four) characters.

An additional problem concerns highly synthetic languages. The more different forms of an individual word that are likely to appear in a text, the greater the effective vocabulary size SRP must use becomes. This dulls the effectiveness of the log transformation, and makes it likely that a rare inflection of a word will be lost in the noise in the corpus. Although this seems as though it should be a major problem, in practice languages like Turkish and Hungarian appear not to suffer greatly.

Finally, SRP assumes consistency in spelling across a corpus. Due to OCR, printing, and spelling errors, this assumption is always somewhat incorrect, but in some cases it is grossly wrong. The LargeVis visualization below makes clear, for example, that Russian is divided into distinct clusters by the orthographic reform around the Russian revolution. The space of the EEBO corpus of historical English texts (http://eebo.chadwyck.com/home) might be dominated by spelling variants rather than linguistically useful terms.

Log transformation

One final adjustment increases the usefulness of random projection for textual features in particular. In practice, random matrices on texts tend to be dominated by the most common words on every dimension. This is undesirable, because it makes the influence of lower-frequency but more content-specific words hard to disentangle. Since negative signs would cause strange effects with low-frequency words, the logarithms are multiplied by and then clipped so no count can be below zero. The net result of this means that extremely common words contribute only moderately more to the final SRP shape of a document than lower frequency words, and that extremely low-frequency words (those appearing less than once per 100,000 words of text in a document) do not contribute to its SRP score at all. This threshold is rather arbitrary: I have chosen it because a typical book in the HathiTrust is, to a first approximation 100,000 words long. Books just under and above 100,000 words of length have the same accuracy rates in classification tasks. An adjustment to the SRP formula above uses the logarithm of frequency rates compared to the total length of document in tokens, L , to scale lengths.

$$SRP(D)_i = \sum_{n=1}^{W} h(w_n)_i * min([0, log(c_i/L * 10^5)])$$

Log transformation of term frequencies is frequently used in information retrieval, and occasionally used in classification tasks.[23] This log transformation

[23] Zafer Erenel and Hakan Altınçay, "Nonlinear Transformation of Term Frequencies for Term Weighting in Text Categorization," *Eng. Appl. Artif. Intell.* 25, no. 7 (October 2012): 1505-14, doi:10.1016/j.engappai.2012.06.013; Jason D. Rennie et al., "Tackling the Poor Assumptions of Naive Bayes Text Classifiers," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, 616-23.

increases the discriminatory power of SRP on classification tasks, at the cost of some comparability across documents of significantly different sizes. (For an example of the increase in power from the log transformation, see the section on a prestige classification benchmark).

Related literature

SRP can be thought of as a particular species of locality sensitive hashing (LSH) that creates features particularly suited for the textual analysis of books and other long documents based on past work in information retrieval. LSH methods navigate a variety of choices: whether to represent documents in Euclidean space or Hamming spaces, and what distance metric in the original textual space to attempt to retain in the new one.[24] Most LSH algorithms are reasonable candidates for a minimal dimensionality reduction; the differences lie in the type of problems that they aim to solve. LSH algorithms have seen use in digital humanities scholarship as part of duplicate detection work. Douglas Duhaime uses hashes across three-letter strings to identify pieces of poetry with close resemblances to each other[25] and Lincoln Mullen includes min-hash in his textreuse library and uses it to detect reprintings.[26]

SRP makes a set of choices and assumptions specifically chosen to be useful on long texts in most human languages. It represents data in Euclidean space because this allows the easiest translation into most widely-used clustering and classification methods; it uses cosine similarity rather than Jaccard similarity on the grounds that frequency becomes an increasingly strong signal in longer texts. It assumes that "words" exist and are, as represented by a simple tokenization algorithm, give better features for study than fixed-length series of bytes; it assumes that lowercasing Unicode characters will usefully combine similar features; and, unlike that a log transformation is a useful step in text pre-processing, since word frequencies tend to follow a power law.

Another similar method is the so-called "hashing trick" widely used in natural language processing. It differs from SRP by hashing each word to a single position

[24] For a good overview, see Jingdong Wang et al., "Hashing for Similarity Search: A Survey," *arXiv:1408.2927 [Cs]*, August 13, 2014.

[25] Douglas Duhaime, "Plagiary Poets. Plagiary Poets," 2016.

[26] Lincoln Mullen, *Textreuse: Detect Text Reuse and Document Similarity*, version 0.1.4, 2016; see also Kellen Funk and Lincoln Mullen, "The Spine of American Law: Digital Text Analysis and U.s. Legal Practice," *American Historical Review* 123, no. 1 (2018).

in the output vector of length N, while SRP places each word into each vector. This creates great advantages in computation, particularly when working with small texts, because a document with only a few dozen texts can be represented and stored with only a few dozen calculations.[27]

Effective use of the hashing trick therefore involves output vectors of quite high dimensionality (Weinberger et al. test in the range of 1 million to 100 million buckets).[28] In the cases where the hashing trick is most frequently used, parsing texts such as e-mails, the sparsity of the output vectors means that the output vectors can be quite small: in a dataset of 2 million usenet posts, each post has on average 165 distinct tokens, which would take 1.3kb to store in sparse form). Books in the Hathi Trust, on the other hand, typically have about 11,000 distinct tokens; an output vector for a single vector in a dimensionality high enough to avoid collisions would take 88 kilobytes. A 640-dimensional SRP, on the other hand, uses only about 2.5 kilobytes per document regardless of the input text's size. SRP is thus better suited to texts that are relatively long, so that a fairly lossy, dense vectorized representation is more effective than a sparse one.

Choosing dimensionality

One notable feature of SRP space is that anyone creating SRP vectors can increase the resolution to an arbitrary level. For certain tasks such as linear language classification reasonably good results can be obtained in as few as ten dimensions; more complicated tasks such multilingual subject classification take at least a few hundred. This paper releases 1280-dimensional SRP vectors for the HathiTrust. This number is chosen because it creates an output file size of about 64 gigabytes; any larger begins to approach the original documents in size. A number of smaller files are also included at resolutions of 50, 320, and 640; the fifty-dimensional vectors, at a little under 3GB, can be loaded into memory on many laptops or downloaded over a wireless connection in a reasonable period of time. Section 3 gives some sense of the tradeoff in using higher or lower dimensionalities in applied tasks.

[27] Kilian Weinberger et al., "Feature Hashing for Large Scale Multitask Learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09 (New York, NY, USA: ACM, 2009), 1113-20, doi:10.1145/1553374.1553516; Qinfeng Shi et al., "Hash Kernels for Structured Data," *Journal of Machine Learning Research* 10, no. Nov (2009): 2615-37.

[28] Weinberger et al., "Feature Hashing for Large Scale Multitask Learning."

Uses of a minimal, universal dimensionality reduction

It may seem implausible that randomly inverting signs on counts of words can produce anything of use for digital humanities research. The rest of this paper, therefore, shows a few of the potential uses of SRP space through examples. These represent a starting point; a reduced-dimensionality space of this sort has several other potential applications in digital humanities research, infrastructure, and pedagogy. The github repository for this paper includes some ipython notebooks sketching out other possible uses.

Overview visualization of the Hathi Trust

Since none of the individual dimensions are meaningful, it is potentially difficult to tell what the relationships among books that are captured in an SRP space might be. Fortunately, dimensionality reduction allows even lower dimensional visualizations of large corpora as one of its major outputs, making it possible to create visual bibliographic maps of any textual collection.

I include one of this bibliographies here: a large-scale map of the millions of books in the Hathi Trust digital library, where books are arranged using only textual features.[29] LargeVis, a technique for visualizing high-dimensional spaces, provides an especially illuminating two-dimensional view of the SRP space. LargeVis (like the related algorithm T-SNE, which does not scale well to collections of this size) creates 2-dimensional arrangements of points where local clusters retain their coherence. The x and y axes are arbitrary, but at both large and small scales the algorithm tries to position groups of similar documents near to each other. While this process is necessarily imperfect, it gives a partial sense of what kinds of textual features exist in the space that SRP creates.[30] The clustering is created solely with SRP features on the books' full

[29] Similar maps exist of scientific research using network placement algorithms-e.g., Matthew Richardson et al., "The Fundamental Interconnectedness of All Things. Places & Spaces: Mapping Science. Courtesy of Elsevier Ltd. In '8th Iteration (2012): Science Maps for Kids,' Places & Spaces: Mapping Science, Edited by Katy Börner and Michael J. Stamper," 2012, and http://paperscape.org/- but they rely on citation metrics.

[30] Jian Tang et al., "Visualizing Large-Scale and High-Dimensional Data," *arXiv:1602.00370 [Cs]*, 2016, 287-97, doi:10.1145/2872427.2883041. A good description for non-specialists of the uses and abuses of T-SNE is Martin Wattenberg, Fernanda Viégas, and Ian Johnson, "How to Use T-SNE Effectively," *Distill* 1, no. 10 (October 13, 2016): e2, doi:10.23915/distill.00002. Another useful method

text; bibliographic information is then overlaid with color to explain or validate the unsupervised clustering.[31]

along similar lines that may work slightly better than LargeVis for capturing large-scale structure is Leland McInnes and John Healy, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv:1802.03426 [Cs, Stat]*, February 9, 2018, http://arxiv.org/abs/1802.03426.

[31] The 1280-dimensional SRP projection of Hathi was projected down to 100 dimensions using principal components analysis; that 100-dimensional space was then stepped down to two dimensions using LargeVis. The PCA step was introduced because it would require expensive hardware to compute LargeVis on a 1280 by 13 million matrix. Unfortunately, it likely also means the clustering captures little information aside from for English or other more common languages.

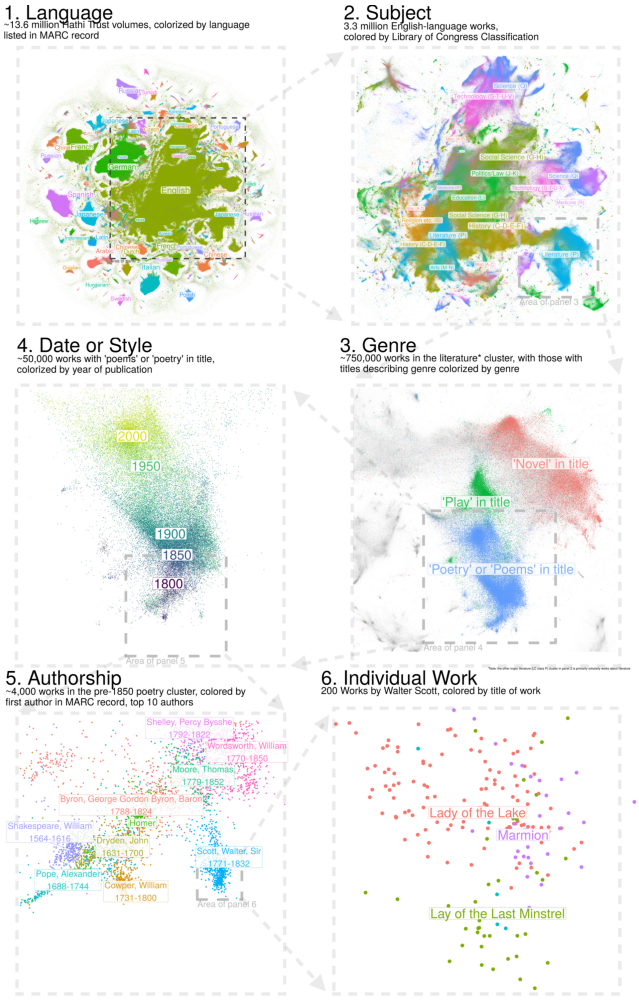

Figure 1. Six successive zoom levels of a single LargeVis dimensionality reduction of the full Hathi set illustrates how a random projection of the Hathi Trust can capture many different types of textual similarity and difference.

The static image above shows six successive zoom levels of this single reduction. At the farthest approach (Panel 1), the visualization is separated by language with the largest language, English, at the center. Languages are relatively distinct, although there are a number of subclusters that include, for example, bilingual works.

Panel 2 shows Library of Congress classification headings for the English-language cluster in panel 1. While languages tend to segregate apart, classes blur together at their edges. The social sciences and education (H, L) occupy the central position; at the top they shade into, first, technology (T) and then the physical sciences and mathematics (Q). To the right they blur into a peninsula occupied first by agriculture (S), then a second cluster of science (Q) containing mostly biological science, which finally ends with a promontory of medical texts (R). A secondary cluster of class R located among education and psychology; it relates more closely to nursing and patient care, while the cluster to the right embraces more pathology and medicine. The south contains the humanities; histories of various regions are intermingled in a way that does not respect the Library of Congress's strict division between the Americas (E and F) and the old world (D), while literature (P) forms a coherent region in the southeast. Music, art and bibliography are clustered together in the south, near a second literature cluster that includes criticism and literary history.

Panel 3 shows a small portion of that overall library: literature written in or translated to the English language. Library metadata does not distinguish well between poems, poetry, and plays, but many works have one of those terms in their title. Using title keywords as a color key shows that the clustering segregates works by genre.

The last three panels show some features of the organization of the literature cluster. (Panel 4) Within one genre, poetry, the overall organization is predominantly chronological, with poems published in the last fifty years closer to the prose genres. (Panel 5) Closer examination of a small portion of the poetry cluster reveals it to segregate individual authors from each other; and (Panel 6) within a single author, Walter Scott, different regions are occupied by distinct individual works.

Each of these levels of organization may have research uses of its own. For example, the distinctions between different copies of the same work (which exists at better precision in the high-dimensional space than in the general-purpose reduction here) may be useful for tasks like detecting duplicates within a corpus, or finding works in Hathi that appear identical to those in another corpus (such as, for instance, Project Gutenberg, which lacks much library metadata). Existing bibliographical information makes duplicate detection and corpus alignment quite difficult; features like these may be useful in facilitating reconciliation into higher-level works.[32]

Other regions of the overall chart show similar macro-micro organization. In or-

[32] Karen Coyle, "FRBR, Twenty Years on," *Cataloging & Classification Quarterly* 53, no. 3 (May 19, 2015): 265-85, doi:10.1080/01639374.2014.943446.

der to make the full Hathi collections browsable in a single image, I have designed a zoomable visualization (Interactive 1 that loads additional books in focused-on regions using the same principles as web mapping tiles. It is impractical to visualize the entire Hathi Trust collection at once. (There are more volumes in the collection than pixels on a typical computer monitor.) Hovering over a point with the mouse displays basic bibliographic information, and clicking links to the volume's full page on hathitrust.org. By choosing any arbitrary point and zooming in, the reader can see what kinds of volumes are present; interactive controls make it possible to filter by subject, date, and title metadata.

This visualization can serve as a kind of guide to some of types of textual attributes that the SRP dataset can be used to analyze. If a cluster is coherent in the visualization, then it also exists in some sense in the higher-dimensional SRP space; relations that do *not* exist in the visualization may exist in the underlying data, or may be lost. At the same time it shows how even a minimal dimensionality reduction enables synoptic views of extremely large corpora; without dimensionality reduction, it would be all but impossible to create any meaningful arrangement of a digital library of this size.

Pairwise and groupwise similarities

In addition to enabling overview exploration and visualization, reduced features can be used in a wide variety of tasks involving the comparison of individual texts to each other. These features can at once work at the smallest scales of similarity-such as identifying duplicate works inside a corpus-and at larger scales-such as finding works similar to a seed text or texts. These similarities are particularly effective in cases where existing metadata is incomplete, inconsistent, or incorrect.

Identification of Duplicate Works

These features are effective at tasks like duplicate detection and work reconciliation as they actually happen in digital libraries. Duplicate detection can be a poorly defined problem in digital library research: different editions of the same work, for example, may or may not count as duplicates, and older editions frequently bind multiple works within the same covers.[33]

[33] Coyle, "FRBR, Twenty Years on."

Still, a simple heuristic suffices to identify duplicates in the Hathi Trust library. As an example, take the set of all books by Charles Dickens in the corpus. There are 2,774 English-language books identifiable in the dataset here identifiable as written by Dickens; 1174 are identifiable based on title metadata as containing one or more of 19 distinct works by Dickens.[34]

In the scheme used here, any pair of books can have one of four relationships to each other:

1. Different titles;
2. The same title and no volume-identifying information;
3. The same title, but different volume information;
4. The same title and identical volume-identifying information. (This does not mean that they contain exactly the same text; one publisher might split "David Copperfield" into 3 volumes, while another might split it into two).

The chart below shows the relationship of books by these categories across a variety of SRP distances.

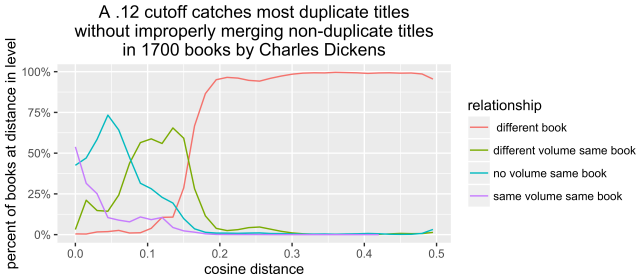

Figure 2. Error in Dickens.

A cutoff of 0.1 or so for cosine distance does quite well in separating class 1 and class 4 from each other that it exceeds, in several cases, the potential of library catalog records. This method, for instance, makes it possible to easily determine the precise novel included in dozens of books identified in library catalogs only through a title such as "Works, Vol. 6." It also reveals cases where the ground

[34] The titles included in this are: *A Child's History of England*, *A Christmas Carol*, *A tale of two cities*, *American Notes*, *Barnaby Rudge*, *Bleak House*, *David Copperfield*, *Dombey and Son*, *Edwin Drood*, *Great Expectations*, *Hard Times*, *Little Dorrit*, *Martin Chuzzlewit*, *Nicholas Nickelby*, *Oliver Twist*, *Our Mutual Friend*, *Sketches by Boz*, *The Old Curiosity Shop*, *The Pickwick Papers.* Titles were identified as belonging to one of these books by virtue of having relevant strings in them: for instance, "nickleby" or "nickelby" to identify copies of Nicholas Nickleby. The rest are miscellaneous other works, or books identifiable only through titles such as "Works - v6."

truth data is actually incorrect: the Hathi volume with id nyp.33433076084767 is improperly labeled in the Hathi catalog as *Hard Times*, even though the title page clearly identifies it as *Little Dorrit*.

Corpus alignment

Corpus alignment is a similar task to duplicate detection that can also be easily executed with SRP features. Corpus alignment is a rarer task than duplicate detection, and is more often practiced using bibliographic identifiers than full text.[35] Without information such as ISBNS, it can be quite difficult to-for instance-identify a hand-corrected Project Gutenberg edition for any given text in the Hathi Trust. Alignment enables the sharing of metadata across corpora.

An examplary task, shown in the supplemental materials, is finding copies in the Hathi Trust of each of the 450 novels in the McGill txtlab's 450 novel corpus in English, French and German. Using a distance cutoff of 0.1 cosine distance successfully matches 377 of 450 novels from the textlab to copies in Hathi with no errors (100% precision, 83.8% recall). In this particular case, a more aggressive cutoff of 0.17 cosine distance correctly matches 405 novels with no errors (100% precision, 91.1% recall).

The precision/recall statistics here measure the Hathi Trust as a corpus, not just the method of SRP. This is important because these are the conditions under which humanists operate: but it also makes it hard to tell the source of errors. It may be that optical character recognition in the Hathi collection is poor, that books do not exist in the Hathi collection at all, or that they only exist bound into multi-volume works. Near-matches occur not because the hashing function erroneously places two unrelated works in the same space, but because the underlying unigram counts are not sufficient to link. For instance, the only copy of Frances Trollope's 1888 novel *That Unfortunate Marriage* in the Hathi Trust is divided into three separate volumes: rather than any of those three, the closest volume in Hathi with a cosine distance of 0.19 is a British novel of four years later (Florence Maryat's 1892 *How Like a Woman*).

As with the Dickens works, this performance is strong enough to reveal places that the existing metadata is incorrect. Some of these are fairly consequential. The metadata to the Txtlab collection identifies a book as Rachilde's *Nono,* when

[35] See, for example, Baumann, Ryan, *Book Aligner* (Web Resource): http://ryanfb.github.io/book-aligner/. Accessed April 27, 2018.

matching algorithms and inspection reveal the text is actually her *Monsieur Venus*. The Hathi collection describes a book as Adele Schopenhauer's *Haus, Wald, und Feldmaerchen* that in facts binds the 350-page novel *Anna* into the same volume as the tales. Others are minor misspellings that would foil many matching algorithms.[36]

Classification

The method and data distributed here are especially suited to bridging classification tasks across multiple languages. Library metadata and journal information give extremely useful information about text corpora, from what disciplines they come from, to the geographical regions they describe.

Prestige classification benchmark

The universal features here compare well to those that humanists typically work with custom-derived for a single set. As an example, take a typical classification task from work by Ted Underwood and Jordan Sellars: distinguishing high- and low-prestige volumes in 19th century poetry.[37] Underwood and Sellars wish to predict whether a volume of poetry will be reviewed, using a model with 3,200 most common words in their corpus of 720 works of English-language poetry. They report 77.5% percent accuracy for a model trained without additional information about year of publication, and 79.2% for a model with year-of-publication information.

Comparing SRP features to the top-N features that Underwood and Sellars use gives a straightforward accounting of how SRP compares to the top-N features widely used in the field. I reran their code using SRP features instead of top-n words as features.[38] With a basic SRP feature set of the same size (3200 dimensions), classification accuracy is 72.7% without year information, and 72.5% with

[36] Among others: the book *Jan Vedder's wife* is listed in the metadata *Jan Veeder's Wife*; *Effi Briest* is spelled *Effie Briest*; *The Vicar of Wrexhill* is titled, instead, *The Vicar of Wrexham;* etc.

[37] Ted Underwood and Jordan Sellers, "The Longue Durée of Literary Prestige," *Modern Language Quarterly* 77, no. 3 (September 1, 2016): 321-44, doi:10.1215/00267929-3570634.

[38] The code for this replication is available in a separate repository that forks the one accompanying their paper.

it. The log transformation yields a substantially higher classification accuracy of 78.61% and 79.03%, equivalent to Underwood and Sellars' original accuracy.

The figure below shows the classification accuracy at a variety of dimensionalities for both top-N and SRP features. These carry some implications for the usefulness of SRP features in general digital humanities tasks:

1. As a rough heuristic, that SRP features are about as good for classification purposes as top-n lists of words of the same length, even though SRP features are language- and content-agnostic.
2. The log transformation in SRP substantially increases the usefulness of the method in classification tasks.
3. Passable classification results are possible with as few as 40 dimensions, but more dimensions continually increase accuracy into thousands of dimensions.

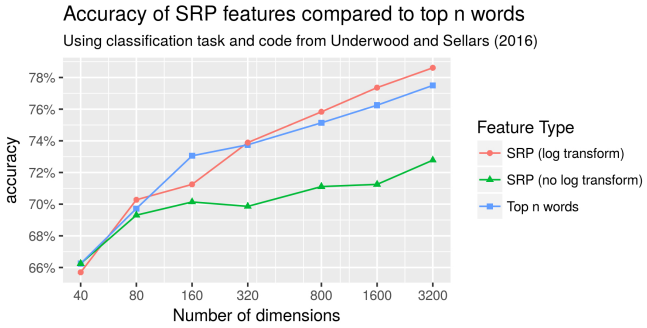

Figure 3. Classification accuracy on Underwood/Sellars dataset

Although the results are equivalent, each of the methods here has useful applications in different frameworks. Top-n features produce more interpretable models (although logistic regressions coefficients themselves are prone to overinterpretation). But SRP features, conversely, may be easier to use in the early stages of a project or if the goal is not to study the classification in its own domain, but to quickly transfer it to a separate set of texts for some other purpose.

Library of Congress Classification

The work by Underwood and Sellars uses logistic regression, in which the core assumption is that features should have linear separability in a space of

words. Although much recent work in the digital humanities has used logistic regression, there are many cases in the digital humanities in which we know the problem should *not* be easily linearly separable. Multilingual classification offers the most obvious instance of this; a high-dimensional space that includes both (say) French and German texts may be easily separable between fiction and non-fiction, but there is no reason to think that a line that separates on French words would work on German words as well, or vice versa.

The particular benefits of SRP features are clear in a rich, multilingual and multiclass problem: attempting to reproduce the Library of Congress classification (hereafter LCC) used to shelve books in many North American research libraries.[39] The classification is hierarchical; at the top level it contains approximately 225 distinct classes, ranging in prevalence within the Hathi collection from 177,000 volumes for the most common class, DS (Asian History), to just 9 for the least common, VD (Naval Seamen). For reference, ten random classes and their counts in the combined training and test sets are shown below.

Training Instances	Class name
461	AI [Periodical] Indexes
6986	BD Speculative philosophy
9311	BJ Ethics
40335	DC [History of] France - Andorra - Monaco
2738	DJ [History of the] Netherlands (Holland)
14928	G GEOGRAPHY. ANTHROPOLOGY. RECREATION [General class]
17353	HN Social history and conditions. Social problems. Social reform
4703	JV Colonies and colonization. Emigration and immigration. International migration
23	KB Religious law in general. Comparative religious law. Jurisprudence
5583	LD [Education:] Individual institutions - United States

Table 1. Ten randomly selected classes from the LCC, with number of occurrences in the corpus.

This classification presents a wide variety of classification challenges that make it useful as a general stand-in for text classification. The texts are multilingual; the classification itself requires extensive expertise to use properly and is not properly reduced to a flat series of buckets as done here. Recent work on reproducing library classifications has tended to include bibliographic metadata as well (occasional) full-text features; they achieve an accuracy between 50% and 75% into more bins than used here deploying bibliographic metadata created along with the classification, such as subject headings.[40] Human-level success rates are un-

[39] https://www.loc.gov/catdir/cpso/lcc.html

[40] Lois Mai Chan, *Cataloging and Classification: An Introduction*, 3 edition (Lanham, Md: Scarecrow Press, 2007) is a useful introduction to catalog practices. Specific assignment tasks, in general using metadata rather than full text, have been attempted on a number of occasions: Ray R. Larson, "Experiments in Automatic Library of Congress Classification," *Journal of the American Society for Information Science* 43, no. 2 (March 1, 1992): 130-48, doi:10.1002/(SICI)1097-

clear: libraries themselves seem to agree in their assignment of LC classification numbers more than 85% of the time, but it is unclear how much of this agreement may be due to cooperative cataloging arrangements.[41]

The classifier is trained on the subset of Hathi volumes that have LCC numbers in their MARC records; this is less than half the full corpus. Additionally, only books (as opposed to serials) are used, since any individual volume of a serial may have a different or more specific subject matter than the full run. (For instance, a general philosophy journal shelved in B might run an issue with only articles about ethics, which is shelved at BJ.) Beyond that those constraints there is no additional filtering: in particular, all languages (including those for which SRP is not especially useful, like Chinese) are included. The full size of corpus is about 3.8 million volumes. 5% of this is randomly set aside as a test set, 90 % is used for training, and 5% is used as a validation set to decide when to halt training.

A variety of training configurations were tested; the final version reported on here was trained using the TensorFlow framework with a neural network using a single hidden layer of 5000 relu nodes. The supporting materials for this paper include code that outlines other minor parameters such as the dropout used in training, and which can be altered create a similar classifier on any metadata field containing either a one-to-one (such as shelf classification, described here) or one to many (such as subject headings).

The results of a number of test classification runs are shown below that illustrates the relative importance of SRP dimensionality and model size. An SRP-based classifier correctly assigns an LC subclass 68% of the time, with 5000 hidden dimensions. A neural network with no hidden layers-which produces the equivalent of a logistic regression model-is a reasonable alternative, classifying correctly just over 60% of the time. Another common nonlinear method, random forests, proved less successful (~58% accuracy). Misses are most often not dramatic; top-three accuracy is 87%. (That is: 87% of the time the actual classification is in the classifier's top three suggestions).

4571(199203)43:2<130::AID-ASI3>3.0.CO;2-S, Eibe Frank and Gordon W. Paynter, "Predicting Library of Congress Classifications from Library of Congress Subject Headings," *Journal of the American Society for Information Science and Technology* 55, no. 3 (February 1, 2004): 214-27, doi:10.1002/asi.10360, and Jun Wang, "An Extensive Study on Automated Dewey Decimal Classification," *Journal of the American Society for Information Science and Technology* 60, no. 11 (November 1, 2009): 2269-86, doi:10.1002/asi.21147. The last has a good bibliography.

[41] Bhagirathi Subrahmanyam, "Library of Congress Classification Numbers: Issues of Consistency and Their Implications for Union Catalogs," *Library Resources & Technical Services* 50, no. 2 (April 2006): 110-19.

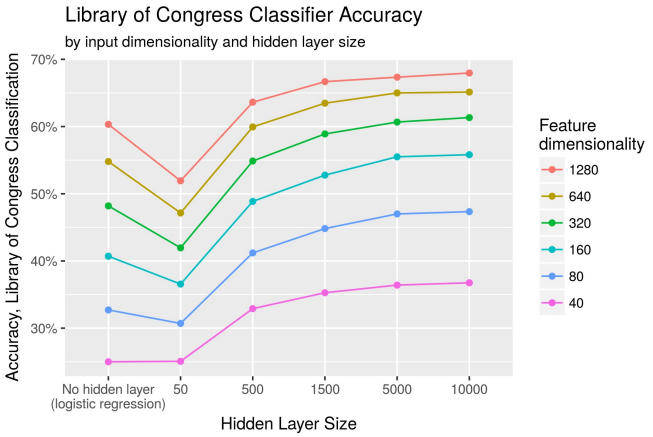

Figure 4. Classifier Success rates by dimensionality and hidden layer size.

Accuracy by language

Since the SRP features preserve linguistic difference, the classifier can run across *all* languages in the Hathi Trust simultaneously. The classifier shows comparable success rates in all of the most common languages in the corpus. Some of its success in less common languages is because a book in, for instance, Polish is likely either Polish literature or Polish history. The accuracy rates for the more non-English languages *excluding* the top two classes are between 30% and 55%; English remains about 67% accurate outside of its top 2. Confirming that a linear classifier exaggerates the advantage of the most common languages, English-language texts are classed at an 8% better success rate than other languages in the linear model, but only 4% better when a hidden layer is introduced. As discussed above, agglutinative languages like Hungarian and Finnish show some of the worst results. Armenian classification seems to be especially poor because of severe shortcomings in Google's OCR for historical Armenian books.

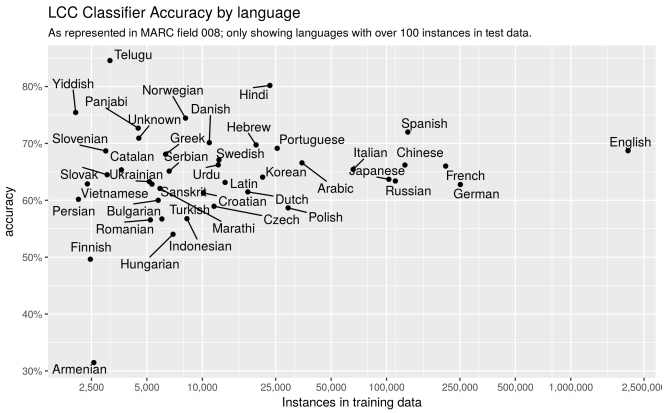

Figure 5. Accuracy by language: one classifier trained on all languages simultaneously.

The historicity of classification

This classifier has many practical uses, but it also suggests the way that classifiers can usefully augment our understanding of the history of library metadata itself.One possible form of interpretation rests in looking at the ways that the classifier fails.

The accuracy of the classifier varies by publication in a striking way way. For the last decades of the nineteenth century, the accuracy of classifier rests above 75%; after 1922, the success rate is only around 68% (and even lower after 1985 or so). This is partly because the composition of the HathiTrust collection changes greatly in 1922, the copyright cutoff in the United States, because some major libraries (including the Harvard University libraries and the New York Public Library) have almost no contributions to Hathi after that date. But when restricting the set to a consistent set of libraries, there is notable evidence of a gradual drop in classifier accuracy that begins in precisely the period when the Library of Congress Classification was created.

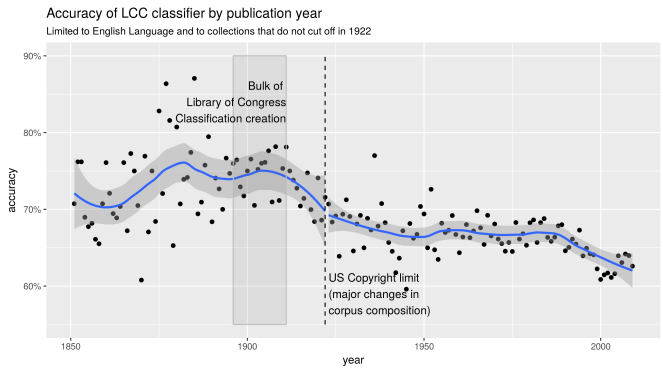

Figure 6. Accuracy by publication date for English language books declines for books published after the creation of the Library of Congress Classification.

This is a correlation which would require much more space to unpack. But it seems possible, at least, that the ontology of the LCC is better suited to books from before 1900 than after; most of the LCC's major divisions were created before 1911, and reflect a division of subject areas that makes more sense in the landscape of late 19th century scholarly production than the present. A classifier trained on the nearly 1000 classes in the Dewey Decimal System, which have been more extensively revised over a longer period of time, does not show a similar drop around 1920.

Reconciling interpretability and accessibility

One way in which SRP features appear not to advance humanistic values is in their interpretability. Logistic models can be interpreted by examining the weights of individual features in the model: neural networks, by contrast, are themselves notoriously hard to inspect. Interpretability is as important virtue for dimensionality reduction as distributability: this is one of the reasons some recent work has begun to use a topic model as dimensionality reduction for supervised and unsupervised tasks.[42]

From a human point of view, though, SRP features can still be used in interpretable ways. Recent work on interpretation of neural networks has suggested

[42] Schoech: Topic Modeling Genre: An Exploration of French Classical and Enlightenment Drama. Pre-print. What Made the Front Page in the 19th Century?: Computationally Classifying Genre in 'Viral Texts'; Jonathan D. Fitzgerald.

that one useful path towards understanding the operation of a network is to progressively disable input features and see how they impact the eventual weights.[43] By examining the *decisions* of a classifier rather than its weights, we can start to understand how and why it works. As an example, take Herman Melville's novel *Moby Dick*. The softmax activations for the book correctly place it in class PS, American literature; the three other classes receiving over a 1% probability are fairly reasonable as well.

Class	Probability
PS American literature	62.70%
PZ Fiction and juvenile belles lettres	30.70%
G GEOGRAPHY. ANTHROPOLOGY. RECREATION	5.40%
PR English literature	1.10%

Table 2. Top predicted classes for Moby Dick, with softmax probabilities.

Moby Dick has about 17,000 distinct word forms by the SRP tokenization scheme. Each can be removed in turn to see how they contribute to the overall weights; and the resulting changes in classification compared to see how each word contributes to the final result. For instance, the following words make the largest difference in terms of how likely it is *Moby Dick* is classed as British literature, relative to the other classes. So, for instance, if all occurrences of the word "American" were removed, the probability of being classes as "PR" would increase from 1.07% to 1.34%; although words are opaque in the SRP features, they remain visible in their impact on the model.

Top positive for class PR	Top negative for class PR
0.300% out (538.0x)	-0.294% as (1741.0x)
0.289% may (240.0x)	-0.292% air (143.0x)
0.258% an (596.0x)	-0.277% american (34.0x)
0.239% had (779.0x)	-0.277% its (376.0x)
0.238% are (598.0x)	-0.250% cried (155.0x)
0.226% at (1319.0x)	-0.246% right (151.0x)
0.221% english (49.0x)	-0.241% i (2127.0x)
0.210% till (122.0x)	-0.231% days (82.0x)
0.209% blow (26.0x)	-0.227% around (38.0x)
0.208% upon (566.0x)	-0.205% back (164.0x)

Table 3. Weights showing terms heavily affecting an SRP-based model decision whether to Moby Dick as British literature (PR).

The supplemental materials include an IPython notebook detailing these statistics for a number of other texts and models. It also includes an example of a

[43] Jiwei Li, Will Monroe, and Dan Jurafsky, "Understanding Neural Networks Through Representation Erasure," *arXiv:1612.08220 [Cs]*, December 24, 2016; Ákos Kádár, Grzegorz Chrupała, and Afra Alishahi, "Representation of Linguistic Form and Function in Recurrent Neural Networks," *arXiv:1602.08952 [Cs]*, February 29, 2016.

true out-of-domain test of the LCC classifier that attempts to determine the plausibility of LCC subclasses assigned to the featured English-language Wikipedia articles from the month of May 2017. The accuracy on Wikipedia is about 50%, with many failures coming in articles about topics like computer games and films which are underrepresented in library books compared to Wikipedia. Accuracy on German-language Wikipedia articles seems to be considerably lower (~25%).

Since creating SRP hashes for a document requires only a small amount of code, it is possible to deploy an in-browser version of the neural network using a pure javascript implementation with no server-side software. This makes it possible to run inference on any arbitrary pasted text entirely on the client side. This version, available online as Interactive 2, includes both the ability to infer classes for any text at all, and to run multiple versions with dropped-out words to see how individual words affect the classification. It also includes a number of other classification models, including one of the top level Dewey Decimal Classification (classes 1 to 999), with 54.3% accuracy.

Conclusion: Research infrastructure

As noted in the introduction, many of the tasks outline in this final section could be accomplished with any sort of infrastructure. The dimensionality reductions that feed into similarity and classification tasks are closely related to the pre-trained *embeddings* created by artificial neural networks. As machine learning becomes more prevalent in the study of cultural artifacts, we are starting to see the distribution of pre-trained models become widespread. For example, the widely used Python module SpaCy uses a single GloVe embedding of words in the English language as the basis of its document similarity scores; and when Google distributed a dataset of 8 million YouTube videos, it released no actual video or images, but instead vectorized features of individual image frames using an image-based neural network.[44]

Such features essentially complement researchers' desire for **machine-readable** texts by offering something new and radically interesting: **machine-read** texts, which offer an abstracted representation of a text based on post-processing by a computer algorithm. It seems possible that existing vectorization or embedding

[44] https://spacy.io/models/en#en_vectors_web_lg; Sami Abu-El-Haija et al., "YouTube-8M: A Large-Scale Video Classification Benchmark," *arXiv:1609.08675 [Cs]*, September 27, 2016.

techniques for documents[45] will eventually expand to the point where vectorized representations of full books offer usefully comprehensive accounts of their contents for selection and research. Elements of a vectorized book interface based on sentence-level embeddings have recently been introduced as part of Google's "Talk to Books" project.[46] The applications of SRP described in the third section, and others possible from the same data, provide a useful point of reference for what kinds of baseline performance we might expect from more exotic approaches, and makes it possible to begin deploying vectorized representations of books inside neural network architectures immediately.

But even if the embedding moment in machine learning eventually sputters out, widely distributed vectorized representations of digital libraries could have a wide variety of uses in the digital humanities. By abstracting out technical aspects of data preparation, they can enable students and beginners to more quickly begin to explore the high-dimensional space of texts. A standardized set of features could have benefits for reproducibility, as well, by making the significance of classification results between studies more immediately comparable.

The classification and visualization examples above illustrate only a few of the ways they let us continue the work of understanding and contextualizing the massive digital libraries that have been created in the past two decades. Preliminary work underway suggests that they can effectively identify misdated works. They can effectively bootstrap out from smaller classifications to assist in the creation of large (tens of thousands of books) custom corpora out of the full digital library. And they make it possible for scholars to search for texts not by individual keywords, but by wholesale semantic similarity, which may offer useful new forms of document discovery.

[45] E.g., Ryan Kiros et al., "Skip-Thought Vectors," *arXiv:1506.06726 [Cs]*, June 22, 2015; Quoc V. Le and Tomas Mikolov, "Distributed Representations of Sentences and Documents," *arXiv:1405.4053 [Cs]*, May 16, 2014.

[46] Daniel Cer et al., "Universal Sentence Encoder," *arXiv:1803.11175 [Cs]*, March 29, 2018. Website at "Talk to Books." Accessed May 11, 2018.